Section I. Examples: For each of the following problems give examples of the languages $J$, $K$, $L$ that are described in the problems. In each case briefly say why the example is a correct one for the stated problem. Each problem is worth 5 points

1. Give examples of two languages $J$ and $K$ both of which are NOT decidable but whose union is decidable.

   Answer: Let $J$ = the halting problem $H$ and let $K$ = the complement of the halting problem, $\overline{H}$. Both are undecidable.

   Then $J \cup K = L$ is the set of all pairs $<M,w>$ where $M$ is any legal TM and $w$ is a string of symbols form $M$’s input alphabet (since for any $<M,w>$ either $M(w) \in J$ halts or $M(w)$ does not halt, so $M(w) \in K$.

   The set $L$ is infinite and decidable.

2. Let $I = \{x \mid |x| \text{ is odd}\}$. (So $I$ is decidable.) Give an example of a language $K \subseteq I$ with $K$ undecidable.

   Answer: Start with the halting problem $H$ (or any other undecidable problem we have looked at will do as well). An element of $H$ is of the form $<M,w>$. Let $c$ be a symbol which is not in $H$’s tape alphabet.

   We construct a set $H_{\text{odd}}$ which looks like a copy of $H$ only with every element in the language having an odd length. To do this let $H_{\text{odd}} = \{ <M,w>c <M,w> \mid <M,w> \in H \}$. Clearly all strings in $H_{\text{odd}}$ are of odd length and $H_{\text{odd}}$ is undecidable since $H$ is.

   Now let $K = H_{\text{odd}}$ the desired subset of $I$.

3. Give examples of two languages $J$ and $L$ both of which are NOT decidable but where $J \cap L$ is infinite and decidable.

   Answer: Let $A$ be the accepting problem, and let $J = A \cup \{v\}^*$. (Note: I am assuming here that $v$ is not in the alphabet of $A$, and also that as usual that $A$ itself is made up of elements of the form $<M,w>$ and so no elements of $A$ is a binary string as it has a comma in it for example.

   Now let $L = \overline{A} \cup \{v\}^*$. Where here $\overline{A}$ is the complement of $A$, so it is all pairs $<M,w>$ where $M(w)$ does not accept. Then $J \cap L$ is just $\{v\}^*$ which is infinite and decidable.
4. Give an example of an infinite decidable set $L$ where $L \subseteq \{M \mid M$ is a Turing machine that does not accept any string$\}$

Answer: Let $T$ be a fixed Turing machine which has input alphabet $\{0,1\}^*$ and tape alphabet $\{0,1,B\}^*$ works as follows. On any binary string $x$, $T(x)$ moves right staying in its start state and goes into its rejecting state. So clearly Turing machine $T$ that does not accept any string.

Not for each integer $i = 1,2,3,...$, let $TM_{T_i}$ be the TM $T$ but with tape alphabet $\{0,1,2,...,i+1,B\}^*$, but otherwise the same as $T$. So in particular $T_i$ works just like $T$ and does not accept any strings. Also, since each $T_i$ has a different tape alphabet each is a different TM. Now let $L = \{T_i | i \in N \}$. Clearly $L$ is the desired infinite decidable subset.

Section II: For each of the following problems 5,6,7 use a reduction to prove that the language $J$, $K$ or $L$ is undecidable.

In each case you must;

- State precisely what problem you are reducing to the problem $J$, $K$, or $L$ and also state precisely what you need to show to establish the reduction, (2 points each), and then
- Give the reduction (4 points each).

5. Show that it is not decidable to determine if a TM $M$ on input $w$ ever goes into the specific state $q_9$ during the computation of $M(w)$. That is, show $J=\{ < M, w > \mid$ the TM $M$ on input $w$ goes into state $q_9$ during its computation $\}$ is undecidable.

Answer: The accepting problem $A$ will be reduced to problem $J$. (That is, we prove $A \leq J$.) Since $A$ is undecidable this implies that $J$ is too.

Recall that $A =\{ < M, w > \mid M(w)$ accepts $\}$.

To show this reduction we need to start with a pair $< M, w >$, an input to $H$ and produce a pair $< T, w >$ with the property that $< M, w > \in H$ if and only if $< T, w > \in J$. Note that the input $w$ is the same for both $M$ and $T$.

Reduction: Our reduction $f$ will just be $f(< M, w >) = < T, w >$, so the function $f$ has to construct $T$ from $M$.

The idea for the reduction $f$ is that we want to construct $T(w)$ so that for any $w$, $T$ goes into state $q_9$ at some time during its computation if and only if $M(w)$ accepts.

This is quite easy to arrange as we just let $T(w)$ be a TM that exactly simulates $M$ on input $w$ but whenever $M$ would accept, $T$ first goes into the specific state $q_9$ and then accepts. In $T$’s program we basically add a transition that causes $T$ to go into $q_9$ before any transition of $M$ that goes into $q_a$.

One final detail. We also need to ensure that the ONLY time $T$ goes into state $q_9$ is when $M(w)$ accepts, that is goes into state $q_a$. To do this we remove the state $q_9$ from $M$’s program replacing it with a new state not originally in $M$. This new state replaces $M$’s state $q_9$ in $T$’s program.
So the result is as we needed, T goes into state $q_9$ at some time during its computation if and only if M goes into state $q_a$ and so accepts.

6. Let $K= \{ M \mid M$ is a Turing machine with input alphabet $\Sigma$ and M accepts every string in $\Sigma^* \}$. Prove K is undecidable.

Answer: We will reduce the accepting problem A to K.

To do this we need to produce a computable function $F$ such that $< M, w >$ is in A if and only if $f(< M, w >)$ is a Turing machine T such that T accepts every one of its input strings.

Reduction.

The idea for the reduction $f$ is that we want to construct T so that for any input $z$, T accepts $z$ if and only if M accepts the specific string $w$.

This is quite easy to arrange as we just let T be a TM which depends on the machines M and input w to M. T, on any that on any input $z$, ignores $z$ and overwrites $z$ with $w$, putting $w$ on its tape. T then simulates M on the input $w$ and does whatever M does on $w$, either accepting, rejecting or never halting.

So in fact T either accepts every one of its inputs, rejects them all or never halts on any of them, depending on what $M(w)$ does.

And T accepts all its input if and only if $M(w)$ accepts. So if we could decide if T accepts every input then we could decide if $M(w)$ accepts, and the reductions works as needed.

7. Let $L= \{ M \mid M$ is a Turing machine with input alphabet $\{0,1\}$ and M accepts the string 001\}. Prove L is undecidable.

Answer: We reduce the accepting problem A to problem J.

To do this we need to produce a computable function $F$ such that $< M, w >$ is in A if and only if $f(< M, w >)$ is a Turing machine T such that T accepts the string 001.

Reduction (in brief). Given M and w, we construct a TM T with input alphabet $\{0,1\}$ as follows.

The idea for the reduction $f$ is that we want to construct T so that T accepts the string 001 if and only if $M(w)$ accepts.

This is quite easy to arrange, similar to the reduction of the previous problem. In fact we can use the same reduction and the same machine T, however assign 0 and 1 to T’s input alphabet if it these two symbols aren’t already in this alphabet.

If $< M, w >$ accepts them T accepts every input string over its input alphabet, and specifically T accepts the string 001. If $< M, w >$ does not accept (rejects or loops) then T does so on every one of its inputs, including 001.

So the resulting reduction works as needed, that is $M(w)$ accepts if and only if T accepts 001.